

Stable Sensor Network (SSN): A Dynamic Clustering Technique for Maximizing Stability in Wireless Sensor Networks

A. B. M. Alim Al Islam¹, Chowdhury Sayeed Hyder², Humayun Kabir³, Mahmuda Naznin³

¹Purdue University, West Lafayette, Indiana, USA

²Michigan State University, Michigan, USA

³Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

E-mail: razi_bd@yahoo.com, hydercho@msu.edu,

mhkabir@cse.buet.ac.bd, mahmudanaznin@cse.buet.ac.bd

Received April 26, 2010; revised May 12, 2010; accepted May 15, 2010

Abstract

Stability is one of the major concerns in advancement of Wireless Sensor Networks (WSN). A number of applications of WSN require guaranteed sensing, coverage and connectivity throughout its operational period. Death of the first node might cause instability in the network. Therefore, all of the sensor nodes in the network must be alive to achieve the goal during that period. One of the major obstacles to ensure these phenomena is unbalanced energy consumption rate. Different techniques have already been proposed to improve energy consumption rate such as clustering, efficient routing, and data aggregation. However, most of them do not consider the balanced energy consumption rate which is required to improve network stability. In this paper, we present a novel technique, Stable Sensor Network (SSN) to achieve balanced energy consumption rate using dynamic clustering to guarantee stability in WSN. Our technique is based on LEACH (Low-Energy Adaptive Clustering Hierarchy), which is one of the most widely deployed simple and effective clustering solutions for WSN. We present three heuristics to increase the time before the death of first sensor node in the network. We devise the algorithm of SSN based on those heuristics and also formulate its complete mathematical model. We verify the efficiency of SSN and correctness of the mathematical model by simulation results. Our simulation results show that SSN significantly improves network stability period compared to LEACH and its best variant.

Keywords: Network Stability Period, Clustering, Energy Consumption Rate

1. Introduction

With the emergence of highly dense fabrication technology and low production costs, wireless sensor networks (WSN) prove to be useful in myriad of diversified applications. In a typical WSN application, sensor nodes are scattered in a region from where they collect data to achieve certain goals. Data collection may be continuous, periodic or event based process. WSN must be very stable in some of its applications like security monitoring and motion tracking. Death of only one sensor node may disrupt coverage or connectivity and thus may reduce stability in this sort of applications. Therefore, all of the deployed sensor nodes in WSN must be active during operational lifetime. However, sensor nodes are gener-

ally equipped with one-time batteries and most of the batteries are of low energy. For this reason, each sensor node must efficiently use its available energy in order to improve the lifetime of WSN. Different techniques are used for efficient usage of this low available energy in a sensor node. Clustering is one of these most well-known techniques.

In some related research works, lifetime of WSN is considered as the time required for the last sensor node to die. Some other related research works also consider lifetime of WSN as the time required for half of the deployed sensor nodes to die. However, lifetime can be termed as stability period of WSN when it considers the time required for the first sensor node to die. There is an impact of efficient usage of available energy in a sensor node on stability period of WSN. This period is mainly

controlled by balanced energy consumption rate throughout the network. Therefore, we have to ensure balanced use of the available energy throughout the network along with the efficient use of the available energy in a sensor node to assure stability in WSN. In this paper, we propose a novel technique Stable Sensor Network (SSN), which ensures balanced use of the available energy throughout the network. It also achieves the efficient use of the available energy in a sensor node by exploiting clustering technique.

Clustering is a technique in which deployed sensor nodes are grouped into some clusters. Only one sensor node is solely responsible to communicate to the base station in a cluster. This sensor node is called cluster head and the remaining sensor nodes in the cluster are called followers. The followers collect data and send it to their corresponding cluster heads. The cluster heads aggregate its own data with the data received from its followers. Aggregated data is then sent to a sink to accomplish a specific goal. Cluster heads remain closer to their follower sensor nodes compared to the sink. It takes less energy to transmit data to the cluster head instead of the sink, which allows the sensor nodes to conserve more energy and live longer in WSN.

There are different clustering techniques already established for ad-hoc networks. However, those techniques cannot be directly used in WSN because of the fact that WSN imposes strict requirements on the energy efficiency than that ad-hoc networks do. As a result, many techniques have been proposed for clustering in WSN. Dynamic clustering techniques are more useful for WSN because of the dynamic variation in residual energies of the sensor nodes. LEACH [1] is one of the simple but popular dynamic clustering techniques used in WSN. LEACH rotates cluster headship very effectively among the sensor nodes of a network based only on some locally available information. However, LEACH does not consider the variation in residual energies of the sensor nodes when it selects the cluster heads. There are some already proposed modifications of LEACH to incorporate the variation. We consider this variation in more efficient way in SSN. We also incorporate the balanced use of residual energy in the network with the help of some heuristics in SSN. We formulate mathematical model for SSN. We prove that SSN has a significant improvement in network stability over LEACH and its best variant by simulation results.

In the next section, we briefly describe some related works. Then, we briefly describe the underlying approach of our technique along with its variants in Section 3. We present our novel clustering technique SSN with three heuristics and complete mathematical model in the next section. In Section 5, we evaluate SSN by simulation results. In last two sections, we conclude the paper shedding some lights on our future works.

2. Related Works

Several techniques have already been proposed to improve network lifetime in WSN. Clustering is one of the widely accepted techniques among them. Clustering is also used in wireless ad-hoc networks, mobile ad-hoc networks along with sensor networks. Several clustering techniques have already been introduced for partitioning nodes in these areas. Some of the early clustering techniques are—Hierarchical Clustering [2], Distributed Clustering Algorithm (DCA) [3], Spanning Tree (or BFS Tree) based Clustering [4], Clustering with On-Demand Routing [5], Clustering based on Degree or Lowest Identifier Heuristics [6], and Distributed and Energy-Efficient Clustering [7], Adaptive Power-Aware Clustering [8]. Some of the recently developed clustering techniques are PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [9], Energy Efficient Clustering Routing [10], PEACH (Power Efficient And Adaptive Clustering Hierarchy) [11], Optimal Energy Aware Clustering [12], ACE (Algorithm For Cluster Establishment) [13], HEED (Hybrid Energy-Efficient Distributed Clustering) [14], PADCP (Power Aware Dynamic Clustering Protocol) [15], LEACH (Low-Energy Adaptive Clustering Hierarchy) [1], SEP (Stable Election Protocol) [16], and LEACH with Deterministic Cluster Head Selection [17].

In [9] PEGASIS introduces a near optimal chain-based protocol. Here, each node communicates only with a close neighbor and takes turns transmitting to the base station, thus reducing the amount of energy spent per round. It assumes that all nodes have global knowledge of the network and employ the greedy algorithm. It maps the problem of having close neighbors for all nodes to the traveling salesman problem. PEGASIS is a greedy chain protocol that is near optimal for a data-gathering problem in sensor networks. Greedy approach considers the physical distance only, ignoring the capability of a prospective node on the chain. Hence, a node with a shorter distance but less residual energy may be chosen in the chain and may die quickly.

In [10] a routing algorithm is proposed which combines hierarchical routing and geographical routing. The process of packet forwarding from the source nodes in the target region to the base station consists of two phases—inter-cluster routing and intra-cluster routing. For inter-cluster routing, a greedy algorithm is adopted to forward packets from the cluster heads of the target regions to the base station. For intra-cluster routing, a simple flooding is used to flood the packet inside the cluster when the number of intra-cluster nodes is less than a predetermined threshold. Otherwise, the recursive geographic forwarding approach is used to disseminate the packet inside target cluster, that is, the cluster head divides the target cluster into some sub-regions, creates the same number of new copies of the query packet, and then disseminates these copies to a central node in each sub-reg-

ion. Like [9], it uses greedy algorithm based on the distance only but not on the capability or the residual energy. Although it deals with the optimal forwarding approach the criteria to choose the cluster heads optimally is not clearly explained.

PEACH [11] is a cluster formation technique based on overheard information from the sensor nodes. According to this approach, if a cluster head node becomes an intermediate node of a transmission, it first sets the sink node as its next hop. Then it sets a timer to receive and aggregate multiple packets from the nodes in the cluster set for a pre-specified time. It checks whether the distance between this node and the original destination node is shorter than that of between this node and already selected next hop node. If the distance is shorter, this node joins to the cluster of the original destination node and the next hop of this node is changed to the original destination node. PEACH is an adaptive clustering approach for multi-hop inter-cluster communication. However, it suffers from almost the same limitations of PEGASIS due to the choice of physical propinquity.

Optimal energy aware clustering [12] solves the balanced k -clustering problem optimally, where k signifies the number of master nodes that can be in the network. The algorithm is based on the minimum weight matching. It optimizes the sum of spatial distances between the member sensor nodes and the master nodes in the whole network. It effectively distributes the network load on all the masters and reduces the communication overhead and the energy dissipation. However, this research work does not consider of residual energy level while choosing a node as the master. Hence, the choice of the master or cluster head is far away from the optimal energy efficient distribution of the cluster heads.

ACE [13] is a distributed clustering algorithm which establishes clusters into two phases—spawning and migration. There are several iterations in each phase and the gap between two successive iterations follows uniform distribution. During the spawning phase, new clusters are formed in a self-elective manner. When a node decides to become a cluster head, it will broadcast a message to its neighbors to become its followers. During migration phase, existing clusters are maintained and rearranged, if required. Migration of an existing cluster is controlled by the cluster head. Each cluster head will periodically poll all of its followers to determine which could be the best candidate to elect as a new leader for the cluster. Current cluster head will promote the best candidate as the new cluster head and abdicate itself from its position. ACE results in uniform cluster formation with a packing efficiency close to hexagonal close-packing. However, ACE does not consider the residual energy of the nodes while selecting cluster heads. Hence, the clustering is far away from the optimal energy efficient.

HEED [14] introduces a distributed algorithm considering the residual energy of sensor nodes. It results in

some clusters by uniformly distributing the cluster heads across the network. It periodically selects cluster heads according to a hybrid parameter which consists of a primary parameter, the residual energy of a node, and a secondary parameter, such as propinquity of a node to its neighbors or node degree. HEED converges in $O(I)$ iterations using low messaging overhead and achieves fairly uniform cluster head distribution across the network. However, it chooses the initial percentage of cluster heads randomly. This random choice remains as a severe limitation of this algorithm.

PADCP [15] uses several adaptive schemes like dynamic cluster range, dynamic transmission power and cluster head re-election to form clusters. In this approach, the sensor nodes are assumed to have the same transmission capability and the ability to adjust transmission power in five levels. PADCP has four major phases—neighbor information collection, cluster head election using a cost function, cluster formation using HEED, and cluster head re-election in case of residual energy lower than a pre defined threshold value. The mobility of the sensor nodes is considered in cluster formation. However, it suffers from the same randomly chosen initial probability limitations of HEED as it completely follows HEED algorithm for cluster formation in its third phase. Moreover, there is no suggestion about the optimal weights of the cost function used in cluster head selection and the threshold used in cluster head re-election.

LEACH [1] introduced a simple mechanism for localized coordination and control for cluster set-up and operation. It also introduces the randomized rotation of the cluster heads and the corresponding clusters. However, it does not consider the variation of the initial energy nor the residual energy of sensors during cluster head selection. SEP [16], a LEACH variant, modifies the equation of the threshold. However, it considers two types of nodes only, normal and advanced, instead of many types that can be encountered in the wireless sensor network after a significant amount of time of operation. Deterministic Cluster Head Selection [17], another variant of LEACH also modifies the threshold to accommodate the heterogeneity of residual energy based on some heuristics. LEACH-C, proposed by the same authors of LEACH in [18], is a centralized technique which selects the cluster heads based on their positions. It considers uniform distribution of the cluster heads based on their positions and the average residual energy in the network. They did not consider the relative residual energy in each sensor node. Adaptive Cluster Head Selection [19], a distributed clustering technique based on LEACH, considers the positions but not the relative residual energies of the sensor nodes.

There are a number of different research works that maximize network lifetime other than clustering. Lifetime is defined in a various ways in those works. In [20], functional lifetime is analyzed solving the linear progra-

ms only for simple and regular network topologies. Functional lifetime of a sensor network is defined as the maximum number of times a certain data collection task can be performed without the death of any sensor node. In [21], average network lifetime is maximized for a sensor network which is under physical node destruction by deriving deployment plan. In [22], α -lifetime of a wireless sensor network is maximized. α -lifetime is the time duration during which at least α portion of deployed sensor area is covered. In [23], a mathematical model is devised for sensor network, where data generation events are spatially and temporally independent. Based on the model, it also introduces a routing protocol for optimal average lifetime. In [24], a method is introduced using the k-shortest simple path algorithm and a dynamic programming method rooted in operational rate-distortion (RD) theory to increase the operational lifetime of a multi-hop 802.15.4 wireless sensor networks. In [25], sensor trees with desired properties are constructed from fusion center and then these sensor trees are scheduled to maximize network lifetime. It considers network lifetime as the time passed before the death of first node in the network. Load Balancing Protocol (LBP) [26] makes the number of live sensor nodes as large as possible by the enforcement of load balancing. Deterministic Energy-Efficient Protocol for Sensing (DEEPS) [27] allows higher energy consumption for sensors with higher total supply and minimizes energy consumption rate for low energy targets. Deterministic Energy-Efficient Protocol for Adjustable Range Sensing (ADEEPS) [28] is an extension of DEEPS. ADEEPS controls sensing range with the underlying approach of DEEPS. In [29], lifetime as time till the death of first node is improved by real time classifier using ART1 neural network model along with co-operative routing. In [30], network lifetime in terms of the death of first sensor node or the first failure of a transmission in the network is maximized by optimal sensor scheduling. It maps the problem to a stochastic shortest-path multi-armed bandit problem and thus chooses the sensor with the largest *Gittins* index for optimal transmission. In [31], Maximum Lifetime Data Aggregation (MLDA) problem is solved by selecting the best data aggregation tree using integer programming. It considers lifetime as the time during which information from all the sensors can be gathered to the base station. In [32], a combinative measurement is defined based on information utility, communication cost, and energy level. Weights of these factors are self-optimized using autonomic computing. In [33], average lifetime is maximized by reducing energy consumption through the enforcement of disjoint sets of sensor nodes. This approach maps Disjoint Set Cover problem to Maximum Flow Problem and then solves the Maximum Flow Problem by mixed integer programming. In [34], average lifetime is maximized by near optimal routing protocol which performs two shortest path computations to route a message. In [35], average lifetime is

maximized by optimal routing through the formulation of linear programming problem. It considers both communication energy consumption rates and residual energy levels of two end nodes in the computation of link cost. In [36], lifetime of a fault tolerant sensor network in terms of death of first sensor node in the network is maximized by using multipath diversity and erasure codes. SPINDS [37] maximizes lifetime in terms of time till the failure of first Aggregation and Forwarding Node (AFN) in two steps. It formulates joint problem of energy provisioning and relay node placement into a mixed-integer nonlinear programming (MINLP) problem. Then it transforms MINLP problem into linear programming (LP) problem with maintaining all critical points in the search space. In [38], sensing ranges of sensor nodes are considered as adjustable. It finds maximum number of set covers and the sensing ranges of sensor nodes to achieve maximum lifetime in terms of time until BS detects the first failure. MLDR [39] attempts to improve network lifetime based on the death of first sensor node in the network by efficient routing using integer programming. This research work also uses data aggregation. In [40], network lifetime based on the death of first sensor node in the network is improved by distributed optimal routing technique using linear programming and sub-gradient algorithm.

A number of research works [20,25,29-31,36,39,40] attempts to improve network stability period by various techniques like—routing, scheduling, aggregation etc. However, in this paper we attempt to improve the network stability period using clustering as it can serve as a better platform for upper layer functionality such as broadcasting, aggregation etc. Our novel algorithm SSN exploits the underlying method of LEACH due to its wide acceptability. In experimental analysis, we compare SSN with LEACH and its best variant. For this reason, we describe LEACH and its variants in detail in the following section.

3. Leach

LEACH is a self-organizing and adaptive clustering protocol [1]. It dynamically creates clusters in order to distribute the energy load evenly among all of the sensor nodes. This algorithm needs time synchronization. Cluster heads are randomly rotated during each time interval. The resultant cluster heads directly communicate with the base station.

3.1. Mechanism

In LEACH, the lifetime of the network is divided into some discrete, disjoint time intervals. Each interval is again divided into some subintervals or rounds as shown in

Figure 1. Each subinterval begins with an advertisement phase followed by a cluster set up phase. In the advertisement phase, each node independently decides whether to become a cluster head or not. In the cluster set-up phase, the clusters are organized based on the decisions made in the advertisement phase. Then a steady-state phase follows. In this phase, the followers, *i.e.*, the sensor nodes except cluster heads, will send data to the corresponding cluster head. The cluster heads accumulate and compress the received data with its own data. Cluster heads send the compressed data to the base station. In order to minimize cluster establishment overhead, the duration of steady-state phase must be longer than that of cluster set-up phase.

At the very beginning of advertisement phase, each node decides whether it wants to become a cluster head for the current round. This decision is based on the suggested percentage of cluster heads for the network, which is set a priori. This decision also depends on the number of times the node has already been a cluster head. This decision is made by a node n choosing a random number between 0 and 1. If the number is less than a threshold $T(n)$, the node decides to become a cluster head. The threshold is calculated as follows:

$$T(n) = \begin{cases} \frac{P}{1 - P \times \left(r \bmod \frac{1}{P} \right)} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

where,

P = the percentage of nodes that can become cluster heads (e.g., $P = 0.05$);

$1/P$ = the number of subintervals in an interval;

r = the current subinterval;

G = the set of nodes that have not been cluster heads yet in the current interval.

Using this threshold, a node can be a cluster head in any one of $1/P$ subintervals in an interval. At the first subinterval of an interval ($r = 0$), each node has a probability P to become a cluster head. The nodes that are cluster heads in the first subinterval cannot be cluster heads in the next $(1/P - 1)$ subintervals of the same interval. Thus the probability that the remaining nodes are becoming cluster heads is increasing. After the completion of $1/P$ subintervals, a new interval will start and all the nodes are again eligible to become cluster head.

Each node that has chosen itself as a cluster head in the current subinterval, broadcasts an advertisement me-

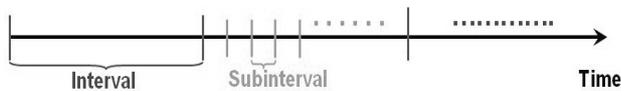


Figure 1. Discrete and disjoint intervals in the whole network lifetime; discrete and disjoint subintervals in an interval.

ssage to the rest of the nodes. The non-cluster-head nodes will choose the cluster to which it will belong in this subinterval. This decision is based on the received signal strength of the advertised message. Assuming symmetric propagation channels, the cluster head whose advertisements have been heard with the largest signal strength will be selected by a non-cluster-head sensor node as its cluster head. In case of a tie, a cluster head is chosen randomly.

3.2. Mathematical Models

There are some incomplete mathematical models available on LEACH. In [18], a mathematical model is proposed to compute the total energy dissipation in the sensor network for the transmission of a frame. Taking the derivative of the total energy it finds the optimum number of clusters, k_{opt} as:

$$k_{opt} = \frac{\sqrt{N}}{\sqrt{2\pi}} \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}} \frac{M}{d_{BS}^2}}$$

where, N is the total number of sensor nodes, M is the dimension of the sensor area, d_{BS} is the distance between cluster head and base station, ε_{fs} and ε_{mp} are the amplifier energies.

In [41], a mathematical model is proposed to compute the total energy consumption in the sensor network during a single round. Taking the derivative of the total energy it also finds the optimum number of clusters, k_{opt} as:

$$k_{opt} = \frac{\sqrt{N}}{\sqrt{\pi}} \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}} \frac{M}{d_{BS}^2}}$$

In [42], a mathematical model is proposed to calculate the total energy consumption in the sensor network during a single round. It also finds the optimum desired cluster head probability, p_{opt} as:

$$p_{opt} = \frac{1}{2} \sqrt{\frac{\varepsilon_{fs}}{\lambda(\varepsilon_{mp} d_{BS}^4 - E_{elec} - E_{DA})}}$$

where, λ is the intensity of homogeneous spatial Poisson process that indicates the sensor node density, E_{elec} is the electronic energy required for coding, modulation, filtering etc. and E_{DA} is the energy required for data aggregation.

However, the lifetime of a sensor node is directly the inverse of its long run rate or expected rate of energy consumption. Therefore, in order to elongate network lifetime, the long run rate of energy consumption must be given more importance than other metrics (for example, energy required to transmit one frame [41] or total energy consumption in an interval [42]). Moreover, none of these models consider the situation in which all the sensor nodes in the network can pick a random number higher than its respective threshold and become a temporary

follower. In this case, no sensor node will find any other node to choose as its cluster head under which it can keep its follower status. In this circumstance, every node changes its state from follower to cluster head, *i.e.*, all the sensor nodes will become a one-member cluster head. In [43,44], a complete mathematical model is proposed incorporating all these factors.

In [48], Heintelman First Order Radio Model [21] is used as the energy model and Renewal Reward Process [26,48] is used as the underlying stochastic process to calculate long run rate of energy consumption. It defines the following parameters in the model:

- 1) P be the desired percentage of cluster heads,
- 2) s be the number of subintervals in an interval, therefore $s = 1/P$,
- 3) P_h be the probability of becoming cluster head of a follower node at the start of any subinterval,
- 4) P_h' be the probability of becoming cluster head of a cluster head node at the start of a subinterval in the next interval,
- 5) Φ_0 be the probability of becoming cluster head of a sensor node at the start of any subinterval,
- 6) $T(n)$ be the currently considered threshold value.
- 7) N be the total number of sensor nodes in the network.
- 8) $a \times b$ be the two dimensions of rectangular sensor area.

According to Renewal Reward Theorem, the rate of reward will be:

$$\lim_{t \rightarrow \infty} \frac{R(t)}{t} = \frac{E(R)}{E(X)} \quad (1)$$

where, R is reward and X is cycle length. It considers the energy consumed by the sensor as the reward and the difference between two consecutive subintervals in which a sensor node becomes cluster head as the cycle.

It considers different state transition diagrams for a sensor node between two states while changing the subinterval in an interval and between two states while changing the subinterval as well as the interval to compute $E(X)$. **Figure 2** shows those state transition diagrams.

Using these state transition diagrams, the probability of becoming a cluster head, Φ_0 , at the start of any subinterval is calculated as follows:

$$\Phi_0 = \left(\frac{P_h}{s}\right) + P_h = (s+1) \times \left(\frac{P_h}{s}\right) \quad (2)$$

After a number of steps, the long run rate of energy consumption is calculated as:

$$\begin{aligned} \lim_{t \rightarrow \infty} \frac{R(t)}{t} &= \frac{E(R)}{E(X)} \\ &= (1 - \Phi_0) \times \left[(E_{elec} k) + (\epsilon_{amp_F} k) \frac{1}{2} \left(\frac{ab}{\pi}\right)^2 \right] \end{aligned}$$

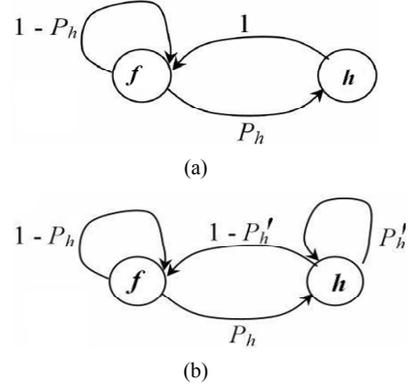


Figure 2. State transition of a node while (a) Changing subinterval without changing Interval, (b) Changing subinterval as well as the interval.

$$\sum_{n=1}^N g(n) \left[\frac{1}{n} - \frac{1}{n+1} \right] + \Phi_0 \sum_{i=0}^{N-1} \left[p(i) \times \sum_{n=1}^{N-i} q(i, n) \right] \quad (3)$$

where,

$$g(n) = \binom{N}{n} \Phi_0^n (1 - \Phi_0)^{N-n} \frac{2n\pi}{ab},$$

$$p(i) = (2i+1)k(E_{elec} + E_{DA}) + (i+1)\epsilon_{amp_H} k\gamma d_{BS}^\lambda,$$

$$\begin{aligned} q(i, n) &= \binom{N-n}{i} h(r, n)^i (1 - h(r, n))^{N-n-i} \\ &\quad \binom{N}{n} \Phi_0^n (1 - \Phi_0)^{N-n}, \end{aligned}$$

and

$$h(r, n) = \frac{\pi r^2}{ab} p_a \left(1 - \frac{\pi r^2}{ab} p_a \right)^{n-1}.$$

Here, E_{elec} is energy required per bit to run the circuitry in transmitter or receiver, E_{DA} is the energy required for data aggregation, ϵ_{amp_F} is the energy constant for the radio transmission of a follower node, ϵ_{amp_H} is the energy constant for the radio transmission of a cluster head node, k is number of bits in a message, λ is the path loss exponent, d_{BS} is the distance between cluster head and base station, and p_a is the percentage of the circular area (centered at a follower and with radius equals distance to a cluster head) falls within the sensor area. As we cannot get any closed form for derivative of Equation 3, we can get the optimal percentage of cluster heads by plotting the value of long run rate of energy consumption from the equation.

3.3. Limitations

This algorithm introduced a fairly simple strategy whi-

ch is more efficient than the direct transmission and the minimum-transmission-energy (MTE) protocol that chooses the route to minimize the transmitter energy. However, it has some limitations:

1) LEACH always wants to achieve an even distribution of energy consumption which might not be rational. Residual energies in different nodes do not remain same after a significant amount of time of operation. Nodes with higher residual energy should get preference to be elected as cluster head. Otherwise, longer network stability as well as longer network lifetime cannot be ensured.

2) When the number of live nodes becomes small, the number of prospective cluster heads which is equal to the number of live nodes multiplied by desired percentage of heads will also become very small and in some cases it may become less than one. For example, if the initial number of sensor nodes is 100 and the desired percentage of heads P is 0.05 then the initial number of prospective heads is $100 \times 0.05 = 5$. However, with the same P when the number of live nodes becomes less than 20 the number of prospective heads will become less than one. Under this condition in most of the subintervals, none of the live sensor nodes can become a cluster head by choosing a random number which is less than the current threshold. In other words, there will be no cluster head available to the sensor nodes to which they can become followers. Rather, all the live sensors will force themselves to become a one member cluster head. In this particular case, the resultant cluster setting will behave like a setting which does not have any clustering. For this reason, no energy efficiency will be gained.

A number of variants have already been proposed for LEACH to overcome its limitations. Some of them are briefly summarized in the following section.

3.4. LEACH Variants

SEP [16] is variant of LEACH, which elects the cluster heads based on weighted probabilities according to the residual energy of the sensor nodes. It assumes that a percentage of the sensor nodes are coming with higher energy resources and studies the impact of heterogeneity of nodes based on their energy levels. It follows the underlying synchronization approach used in LEACH. In addition, it considers the variation in the residual energy assuming two types of nodes—normal and advanced.

It assumes m fractions of the nodes are advanced nodes, which have α times energy than that of the normal nodes. As a result, it assumes $n(1 + \alpha m)$ number of virtual normal nodes in the network. It extends the number of subintervals from $1/P$ to $(1 + \alpha m)/P$ in an interval. The objective of this extension is to elect a normal node once and an advanced node $(1 + \alpha)$ times as the cluster head in an interval. The probability equation to become cluster head has been modified. In fact, two different equations are used for the normal and the advanced nodes.

The weighted election probabilities for the normal and the advanced nodes are p_{nrm} and p_{adv} respectively. Their equations are as follows:

$$p_{nrm} = \frac{P_{opt}}{1 + \alpha \times m}$$

and,

$$p_{adv} = \frac{P_{opt}}{1 + \alpha \times m} \times (1 + \alpha)$$

where, P_{opt} is the optimal probability of a node to become a cluster head. It also uses two different equations for the threshold. One for the normal nodes called $T(s_{nrm})$ and the other for the advanced nodes called $T(s_{adv})$. $T(s_{nrm})$ and $T(s_{adv})$ are calculated as follows:

$$T(s_{nrm}) = \begin{cases} \frac{P_{nrm}}{1 - p_{nrm} \times \left(r \bmod \frac{1}{P_{nrm}} \right)} & \text{if } s_{nrm} \in G' \\ 0 & \text{otherwise} \end{cases}$$

and,

$$T(s_{adv}) = \begin{cases} \frac{P_{adv}}{1 - p_{adv} \times \left(r \bmod \frac{1}{P_{adv}} \right)} & \text{if } s_{adv} \in G'' \\ 0 & \text{otherwise} \end{cases}$$

where, G' is the set of normal nodes that have not become cluster head yet within the last $1/p_{nrm}$ subintervals and G'' is the set of advanced nodes that have not become cluster head yet within the last $1/p_{adv}$ subintervals in an interval.

This works introduced the heterogeneity to LEACH in terms of two levels of residual energy. However, it has some limitations:

- In SEP, the percentage of cluster heads is optimized based on the energy consumption in an interval. However, this value should be optimized on the basis of the long run rate or expected rate of energy consumption for achieving the higher network stability period.
- SEP considers two types of nodes only in terms of residual energy. However, during the life cycle of the network the different levels of the residual energies may exist which will not be covered by only two types. More types of nodes are necessary to consider covering numerous residual energy levels in different nodes to achieve maximum network stability.
- It did not attempt any improvement to enhance the network stability.

Deterministic Cluster Head Selection [17] introduces the heterogeneity to LEACH in terms of residual energy. It considers the residual energies of the sensor nodes in order to manage rational power consumption throughout the network. It follows the underlying mechanism of LEACH exactly. It has changed the equation of the thresh-

old value only to incorporate the residual energy in cluster head selection process as follows:

$$T(n)_{new} = \frac{P}{1 - P \times \left(r \bmod \frac{1}{P} \right)} \times \frac{E_{n_current}}{E_{n_max}}$$

where, $E_{n_current}$ is the current energy, E_{n_max} the initial energy of the node. The other parameters have the same definitions as of LEACH.

After a significant amount of time of operation, the residual energies of the sensors would become very low and then this threshold value will be very low. This can result in a situation where all the live sensors are one member cluster head. In this case the energy consumption rate will be very high. To break this stuck condition another modified equation of the threshold value has been proposed:

$$T(n)_{new} = \frac{P}{1 - P \times \left(r \bmod \frac{1}{P} \right)} \times \left[\frac{E_{n_current}}{E_{n_max}} + \left(r_s \text{div} \frac{1}{P} \right) \times \left(1 - \frac{E_{n_current}}{E_{n_max}} \right) \right]$$

where, r_s is the number of consecutive rounds in which a node has not been cluster head.

This works introduced the heterogeneity to LEACH in terms of different levels residual energy. However, it has some limitations:

- Deterministic Cluster Head Selection uses a random value for the percentage of heads parameter like LEACH. Therefore, it does not consider the optimal value of this parameter.
- It does not suggest any optimum value for r_s .
- It did not attempt any improvement to enhance the network stability.

LEACH-C [18] is a centralized technique to cluster sensor nodes based on their positions. In this approach, base station selects cluster heads to get uniformly distributed clusters. In LEACH-C, sensor nodes detect their current locations using GPS (Global Positioning System) receiver or any other technique. At the beginning of each interval, each node informs the base station its current location and residual energy level. After receiving the information from all the sensor nodes, base station computes the average residual energy in the network. It precludes those sensor nodes whose residual energy is below the average residual energy from attaining cluster heads-hip. Base station selects the cluster heads from the remaining nodes using the simulated annealing algorithm [47]. Base station also selects corresponding followers for the clusters while selecting the clusters and cluster heads, and the base station broadcasts a message into the network informing these selections.

This algorithm minimizes the total sum of squared distances between all the non-cluster-head nodes and the corresponding closest cluster head node. Thus, it minimizes the amount of energy necessary to use to transmit data to the cluster head nodes by the non-cluster-head nodes. However, it suffers from the following limitations:

- In LEACH-C, the base station selects the cluster heads based on their positions and the average residual energy in the network. Like LEACH, the individual residual energy in each sensor node has little impact on the cluster head selection process in LEACH-C. This centralized algorithm also suffers from non-scalability.
- Incorporating GPS receiver or similar device in the sensor nodes increases sensor node cost.
- It did not attempt any improvement to enhance the network stability.

Adaptive Cluster Head Selection 19 assumes that a sensor node knows its distance from another sensor node by observing the signal strengths in the received messages. At first, this approach randomly selects cluster heads following LEACH. Next it reselects the cluster heads considering the distance between each cluster head and the sensor nodes farthest from the cluster heads. The reselection is done in order to distribute the cluster heads uniformly in the network. When a sensor node is selected as a cluster head by LEACH, it broadcasts an advertisement message to all other nodes. Other sensor nodes respond to the broadcast. From the received responses, a cluster head calculates its distance to its farthest follower node and its distance to the nearest cluster head of neighbor clusters. It subtracts the first distance from the later. Three cases may arise as follows:

Case 1: The result is positive.

Case 2: The result is negative.

Case 3: The result is zero.

In order to place the cluster head in an optimum location, the cluster head is moved to the direction of the closest head in Case 1 and to the direction of the farthest sensor node in Case 2. Cluster head position remains the same in Case 3.

This approach ensures uniform distribution of the cluster heads. However, it has the following limitations:

- It completely ignores the relative residual energy of each sensor node in the network while selecting the cluster heads. It also suffers from other LEACH limitations.
- In this work cluster head movement, if necessary, is not clearly defined.
- It did not attempt any improvement to enhance the network stability.

Therefore, none of the research works mentioned in this section makes any attempt to improve network stability. In the next section, we propose a novel technique, Stable Sensor Network (SSN) to improve this metric.

4. Stable Sensor Network (SSN)

In this section, we propose a new algorithm to cluster sensor nodes in a network to improve network stability in terms of death of the first sensor node. We follow the underlying approach of LEACH. In LEACH, each sensor node is given equal chance to get cluster headship and thus its lifetime depends only on its own residual energy. Therefore, a sensor node with low residual energy dies within a short period. However, there may be some other sensor nodes alive after its death. If that sensor node could exploit the residual energies of the live sensor nodes then it would live longer. Therefore, we should use the total residual energy of the network to increase the lifetime of the sensor node which dies first. We present three heuristics to achieve this goal. We illustrate our complete clustering algorithm SSN in details after describing those heuristics. We also adapt the mathematical model derived in [44] for SSN incorporating the heuristics accordingly.

4.1. Heuristics

We propose three heuristics for SSN. First two heuristics basically attempt to use the residual energy of the network in a sensor node. A sensor node needs residual energy status of other sensor nodes in the network for the second heuristic. It can get that information from unacknowledged broadcasts from other sensor nodes. Some of the information will not be available to the sensor node due to unacknowledged broadcasts. Third heuristic attempts to make up this missing information.

Heuristic 1: Energy consumption of a cluster head node is higher than that of a follower node. Therefore, sensor nodes with higher residual energy should be elected as cluster heads. In the original LEACH algorithm, if a node becomes cluster head in a subinterval it cannot become cluster head again in any of the subsequent subintervals of the same interval. However, if a sensor node with higher residual energy can become cluster head again in other subintervals in the same interval then a sensor node with lower energy can escape from being cluster head. In that case, the lifetime of this lower energy sensor node will increase by using residual energy of a higher energy sensor node. For this reason, we make the subintervals completely memory less and eliminate the use of the separate set of nodes that have not been cluster head yet in the current interval. With this modification, the probability of becoming cluster head of a sensor node in a subinterval does not depend on its status in the previous subintervals. This heuristic partially increase network stability period.

Heuristic 2: We can expect higher stability period of a sensor network if we increase the probability to become cluster heads for sensor nodes with higher residual energies. We should consider relative residual energy of a

sensor node to determine whether it is with higher residual energy or not. For this reason, we judge the relative residual energy of a sensor node while selecting it as a cluster head. We map the relative residual energy of a sensor node in its threshold computation so that it keeps its expected value at the optimal percentage of cluster heads P . At the beginning of each subinterval, each node knows its own residual energy (E_{cur}) along with maximum (E_{cur_max}), minimum (E_{cur_min}), and average (E_{cur_avg}) residual energies of all sensor nodes alive in the network. Considering average residual energy (E_{cur_avg}) corresponds to expected percentage of cluster heads (P), we map E_{cur_min} and E_{cur_max} to $(1 - P_{range})$ and $(1 + P_{range})$ respectively, where P_{range} is the minimum between P and $(1 - P)$. If $P \leq (1 - P)$, $(P - P_{range})$ becomes zero and if $P \geq (1 - P)$, $(P + P_{range})$ becomes one. This has been shown in **Figure 3**.

We define deviation from P for a sensor node based on the difference between its residual energy E_{cur} and the average residual energy E_{cur_avg} in the network. Hence, the deviation is:

$$\Delta P = P_{range} \times E_r \quad (4)$$

where,

$$E_r = \begin{cases} \frac{E_{cur} - E_{cur_avg}}{E_{cur_avg} - E_{cur_min}}, & \text{if } E_{cur} < E_{cur_avg} \\ 0, & \text{if } E_{cur} = E_{cur_avg} \\ \frac{E_{cur} - E_{cur_avg}}{E_{cur_max} - E_{cur_avg}}, & \text{if } E_{cur} > E_{cur_avg} \end{cases}$$

In order to make the threshold value proportional to the residual energy of a sensor node, we assign threshold value equal to P plus ΔP , i.e.

$$T(n) = P + \Delta P \quad (5)$$

This heuristic along with the previous one enable a sensor node to become cluster head according to its relative residual energy in the network. A sensor node with higher residual energy is ensured to be more probable and a sensor node with lower residual energy is ensured to be less probable in the selection of cluster heads.

Heuristic 3: To apply the previous heuristic, a sensor node must know the maximum, minimum, and average residual energies of all sensor nodes alive in the network. To calculate these values it must know residual energies

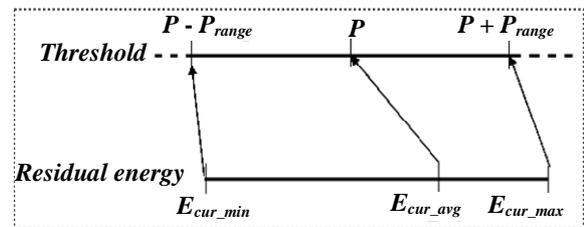


Figure 3. Distribution of Threshold Value according to Residual Energy.

of all other sensor nodes. Therefore, each node must broadcast its residual energy level. For guaranteed availability of this information, some acknowledgement based data transmission technique should be followed. However, this will incur a significant energy cost. For this reason, we make a trade-off between the accuracy of this information and the energy required to transmit them. We simply adopt one pass broadcast and to overcome the accuracy problem we multiply the threshold by the ratio between total number of deployed sensor nodes (N) and number of sensor nodes (N_{live}) from whom residual energy information is received. This will increase the probability of a sensor node to become cluster head when it finds a lower number of live sensor nodes in the network. This ultimately ensures the preservation of overall optimal percentage of cluster heads among those reachable sensor nodes irrespective of the statuses of the unreachable sensor nodes. This heuristic changes the equation of the threshold as the following way:

$$T(n) = (P + \Delta P) \times \frac{N}{N_{live}} \quad (6)$$

4.2. SSN Algorithm

We divide the lifetime of the network into some discrete and disjoint equal length intervals in SSN. Each interval has three consecutive phases—advertisement, cluster-setup, and steady-state phase. The algorithm depicted in **Figure 4** runs independently in each sensor node in each interval. The parameters are initialized at the start of the algorithm. E_{cur} is set to its current residual energy level. E_{cur_max} , E_{cur_min} , and E_{cur_avg} are set to its own current residual energy level, *i.e.*, equal to E_{cur} . The number of live sensor node, N_{live} , is set to one assuming it is the only live sensor node in the network. Advertisement, cluster-setup, and steady-state phases are executed as follows:

1) Advertisement Phase: During this phase, each node executes two parallel processes. In one process, each node waits for a uniformly distributed random amount of time and then broadcasts its current residual energy level. This random delay reduces the probability of collision. Another process receives the current residual energy levels of other sensor nodes. A sensor node may receive multiple copies of a current energy level advertisement message from the same sensor node due to multi-path effect. A receiver sensor node detects these duplicate receptions and ignores them. A receiver sensor node updates the parameters— E_{cur_max} , E_{cur_min} , E_{cur_avg} , and N_{live} using the fresh advertisement messages only.

2) Cluster Set-up Phase: In this phase, each sensor node independently decides whether to become a cluster head or not based on the information gathered in the advertisement phase. At first, it calculates the threshold $T(n)$

using Equation 6. Next, it picks a random number and compares the random number with the threshold. Three cases may arise as follows:

CASE 1: The random number is less than the threshold. In this case, the sensor node becomes a cluster head and broadcasts HEAD_EXPOSURE message.

CASE 2: The random number is not less than the threshold and it does not receive any HEAD_EXPOSURE message from other sensor nodes. In this case, the sensor node becomes a one member cluster head.

CASE 3: The random number is not less than the threshold and it receives one or more HEAD_EXPOSURE messages from other sensor nodes. In this case, the sensor node becomes a follower of the nearest cluster head and sends a FOLLOWER_ACCEPTANCE: Message to the nearest cluster head.

3) *Steady-state Phase*: In this phase, the followers send data to the corresponding cluster head. The cluster heads accumulate, aggregate, and compress the received data with its own data. Cluster heads send the aggregated and compressed data to the base station. The duration of steady-state phase is significantly longer than the summation of the durations of the advertisement and cluster-setup phases in order to minimize cluster establishment overhead.

4.3. Mathematical Model of SSN

The difference between underlying mode of operations of LEACH and SSN arises because of three new heuristics. The last two heuristics make change only in the threshold value ($T(n)$). This change merely affects the probability of becoming cluster head of a follower node at the start of any subinterval (P_h). Otherwise, there is no impact of these two heuristics on Equation 3, which is the latest mathematical formulation of LEACH. However, *heuristic 1* of our new clustering algorithm makes the subinterval completely memory less. For this heuristic, the first state transition diagram of **Figure 2** is no longer applicable. However, Φ_0 is formulated from the weighted combination of two state transition diagrams of **Figure 2** in the mathematical model of LEACH. Therefore, the formulation of Φ_0 needs to be changed in the mathematical model of SSN. With the introduction of *heuristic 1*, any sensor node can become cluster head irrespective of its status in the previous sub interval. Therefore, the probability of becoming cluster head of a follower node at the start of any subinterval (P_h) will no longer differ from the probability of becoming cluster head of a sensor node at the start of any subinterval (Φ_0). As a result, formulation of Φ_0 in the changed mathematical model of SSN will be $\Phi_0 = P_h$. With this change, we can use Equation 3 as the mathematical model of SSN.

<p>Algorithm SSN () Set the value of E_{cur} Initialize E_{cur_max}, E_{cur_min}, and E_{cur_avg} to E_{cur}. Initialize N_{live} to 1 Advertisement(); broadcasts and receives current energy levels Cluster_Set_Up(); generates the clusters Steady_State(); receive and transmit data</p>
<p>Algorithm Advertisement() Transmit_Current_Energy_Residual_Level() Receive_Current_Residual_Energy_Level()</p>
<p>Algorithm Transmit_Current_Residual_Energy_Level() Wait for a random time Broadcast own current residual energy level</p>
<p>Algorithm Receive_Current_Residual_Energy_Level() For each received current residual energy level, E'_{cur} if E'_{cur} is not a repetition from an already received node Update_Parameters(N_{live}, E'_{cur}) endif</p>
<p>Algorithm Update_Parameters(N_{live}, E'_{cur}) if $E'_{cur} > E_{cur_max}$ $E_{cur_max} = E'_{cur}$ endif if $E'_{cur} < E_{cur_min}$ $E_{cur_min} = E'_{cur}$ endif $E_{cur_avg} = \frac{N_{live} * E_{cur_avg} + E'_{cur}}{N_{live} + 1}$ Increment N_{live}</p>
<p>Algorithm Cluster_Set_Up () $P_{range} = \max(P, 1 - P)$ if $E_{cur} < E_{cur_avg}$ $E_r = \frac{E_{cur} - E_{cur_avg}}{E_{cur_avg} - E_{cur_min}}$ else if $E_{cur} > E_{cur_avg}$ $E_r = \frac{E_{cur} - E_{cur_avg}}{E_{cur_max} - E_{cur_avg}}$ else $E_r = 0$ endif $\Delta P = P_{range} \times E_r$ $T(n) = (P + \Delta P) \times \frac{N}{N_{live}}$ Choose a random number r if ($r < T(n)$) then $status = head$ broadcast HEAD_EXPOSURE message else Receive HEAD_EXPOSURE messages from other sensor nodes if (no HEAD_EXPOSURE message received) then $status = head$ else $status = follower$ send FOLLOWER_ACCEPTANCE message to nearest cluster head endif endif endif</p>
<p>Algorithm Steady_State () if $status = follower$ send self originated data to own cluster head else receive messages from own followers aggregate and compress them with own message send to base station endif</p>

Figure 4. Algorithms for Stable Sensor Network (SSN).

We compare these mathematical models from simulation results in the next section. We also analyze the efficiency of SSN in that section.

5. Simulation Results

We conduct our simulation runs on a randomly deployed wireless sensor network. Our simulation program is written in visual C++. In this section, we first describe our network settings along with various parameters used in energy rate calculation. Then, we compare mathematical models of SSN with that of LEACH. Finally, we evaluate network stability in SSN with that of LEACH and its high performance variant.

5.1. Network Settings

We use network settings as shown in **Figure 5** in our simulation runs. The network settings do not make any impractical assumption to simplify the analysis. The settings are as follows:

- The dimension of sensor area is $200 \times 200 \text{ m}^2$.
- Total number of sensor nodes in the network is 100.
- The sensor nodes are randomly distributed over the sensor area.
- Each sensor node is initially equipped with a battery of 5 Joule.
- The base station is located at position (400 meter, 100 meter).

We use the following parameters in the simulation runs for verification of mathematical models and in the first runs of the subsequent analyses:

- 1) The amount of energy per bit to run sensor node circuitry, E_{elec} is 5×10^{-8} ;
- 2) The value of energy constant, C_{amp} , for radio transmission, is 1×10^{-10} ;
- 3) The number of data packets generated during each subinterval by a sensor node is normally distributed in the range of $[0, 50]$, with the value of mean equal to 25. We applied Box-Muller transformation [48] to achieve this normal distribution from the uniform distribution of the built-in *rand()* function in visual C++.

- 4) Each data unit contains 8 bits data;
- 5) The probability that a message successfully arrives at its destination is 90%.

5.2 Verification of Mathematical Model

We first plot the long run rate of energy consumption from Equation 3 versus the percentage of heads for a LEACH node in **Figure 6**. The value of the probability of becoming cluster head of a sensor node at the start of any subinterval Φ_0 must not exceed 1 and the value of Φ_0 can be computed from Equation 2. According to Equation 2, if the value of P exceeds 0.61 then the value of Φ_0 will exceed 1. In order to avoid this, we plot the graph against the percentage of cluster heads P up to 0.61. According to the graph:

- 1) Energy consumption rate initially decreases very sharply with the increase of the percentage of cluster heads.
- 2) There is an optimal point for which energy consumption rate is the lowest. After this point the energy consumption rate increases with the increase of the percentage of cluster heads. In our simulation runs this optimal point is (0.045, 0.000337).

We also plot the long run rate of energy consumption from Equation 3 versus the percentage of heads for a SSN node in **Figure 7**. Here, the probability of becoming cluster head of a sensor node at the start of any subinterval Φ_0 is no longer computed from Equation 2 as in SSN Φ_0 directly maps to P_h . Therefore, we plot the graph against the percentage of cluster heads P up to 1.

The graph in **Figure 7** exhibits almost the same trends found in the graph in **Figure 6**. Energy consumption rate initially decreases very sharply with the increase of the percentage of cluster heads and after an optimal point the energy consumption rate increases with the increase of the percentage of cluster heads. In **Figure 7**, the optimal point for SSN is (0.045, 0.000331) which gives lower long run rate of energy consumption than the optimal point for LEACH found in **Figure 6**. This improvement is only due to *heuristic 1* as only this heuristic modifies the mathematical model.

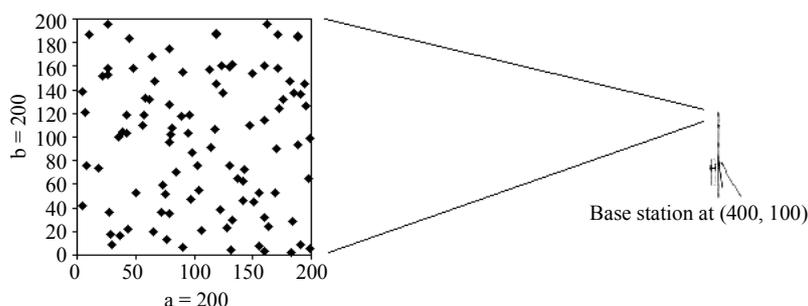


Figure 5. Network Settings: Uniformly distributed sensor nodes with a base station.

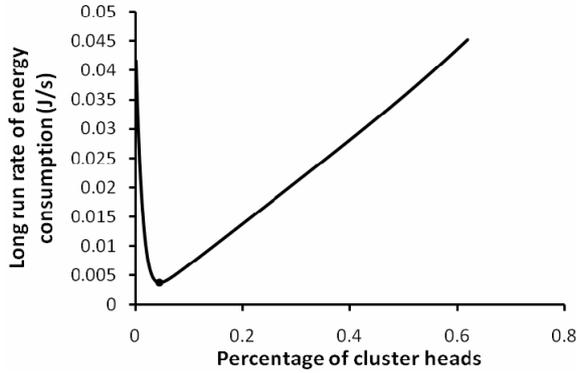


Figure 6. Long run rate of energy consumption against Different Percentage of Heads according to the Mathematical Model of LEACH.

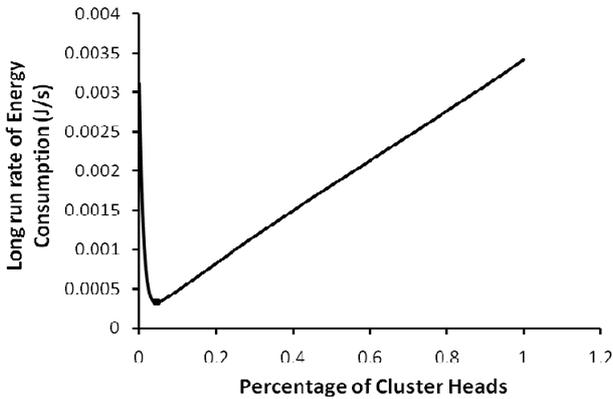


Figure 7. Long run rate of energy consumption against Different Percentage of Heads according to the Mathematical Model of SSN.

We evaluate network stability of SSN against that of LAECH and its best variant. The authors of Deterministic Cluster Head Selection [17] claimed that it improves the network stability period by 30% over LEACH whereas the authors of SEP [16] claimed that it does the improvement over LEACH by 26%. These two are the most improved LEACH variants claimed so far. For this reason, we take Deterministic Cluster Head Selection [17] as the best LEACH variant instead of SEP in our performance comparison. We already find $P = 0.045$ as the optimal percentage of cluster heads from the mathematical models of both LEACH and SSN. We use this value for all of the three techniques under evaluation. Our evaluation is based on three metrics:

- 1) Data rate of a sensor node,
- 2) Position of base station, and
- 3) Initial energy of a sensor node.

In the evaluation process, for each simulation run we take average of values found from fifty simulation passes. Each simulation run generates a point in a graph. We start with already described network settings for the first

point in each of the graphs.

I. Data rate of a sensor node

In the initial network settings, number of data packets generated by a sensor node in a subinterval is normally distributed in the range of 0 to 50, with mean 25. We conduct fifteen simulation runs varying this range. We change the upper limit of the range from 50 packets with step of 10 packets in each simulation run. We plot the values of network stability periods in terms of FND in **Figure 8** and the values of HND in **Figure 9**. **Figure 9** indicates that HNDs of SSN, LEACH and LEACH variant are comparable. However, there is a significant steady improvement in FND for SSN over LEACH and its variant. We plot these improvements in **Figure 10**. The average improvement over LEACH and its variant is 53.42% and 35.62% accordingly.

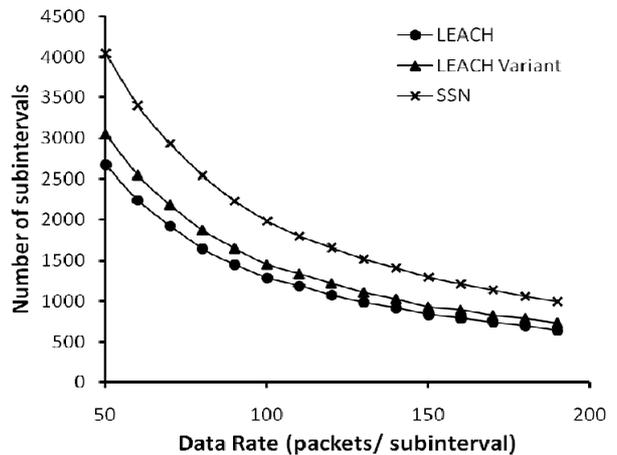


Figure 8. Network stability period in terms of First Node Dies (FND) for different data rates.

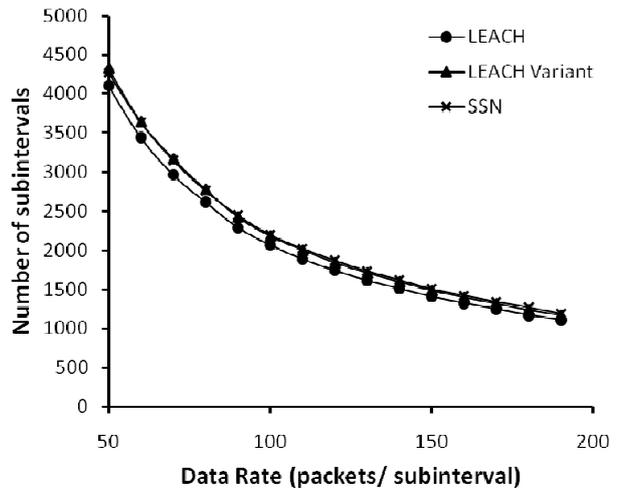


Figure 9. Half of the Nodes Die (HND) for different data rates.

II. Position of base station

In the initial network settings, base station is located at (400 m, 100 m). Therefore, distance of the base station from the center of network area is 300 meter. We conduct fifteen simulation runs varying this distance. We change the position of base station in first dimension from 400 meters with step of 25 meters in each simulation run. We plot the values of network stability periods in terms of FND in **Figure 11** and the values of HND in **Figure 12**. **Figure 12** indicates that HNDs of SSN, LEACH and LEACH variant are comparable. However, there is a significant steady improvement in FND for SSN over LEACH and its variant. We plot these improvements in **Figure 13**. The average improvement over LEACH and its variant is 48.55% and 30.22% accordingly.

III. Initial energy of a sensor node

In the initial network settings, initial energy of a sensor node is 5 Joule. We conduct fifteen simulation runs

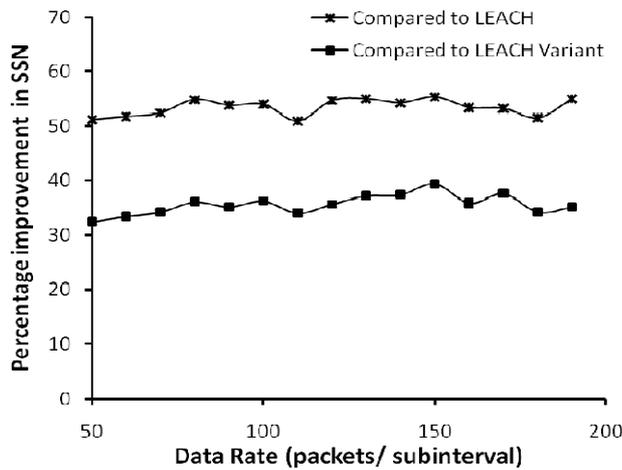


Figure 10. Improvement in network stability period in terms of First Node Dies (FND) in SSN for different data rates.

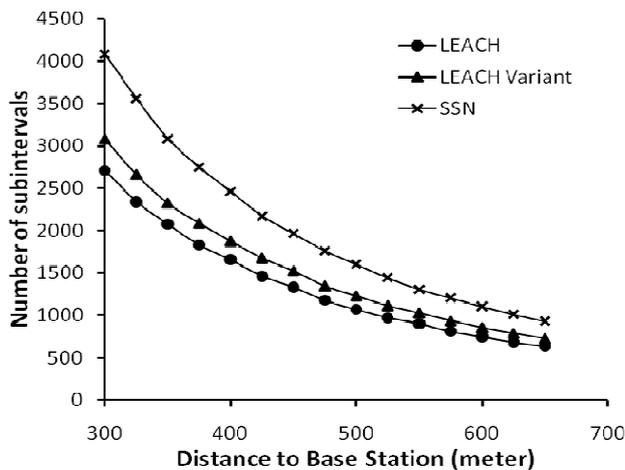


Figure 11. Network stability period in terms of First Node Dies (FND) for different positions of base station.

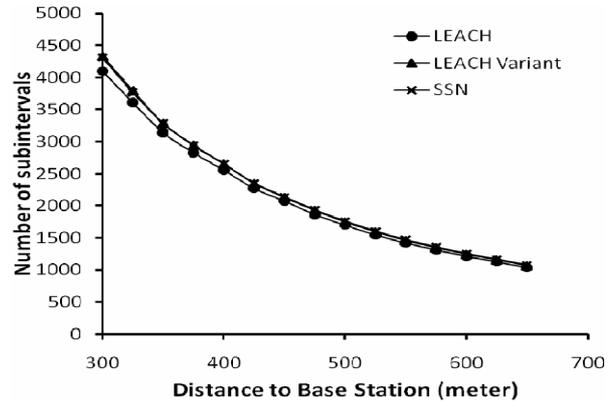


Figure 12. Half of the Nodes Die (HND) for different positions of base station.

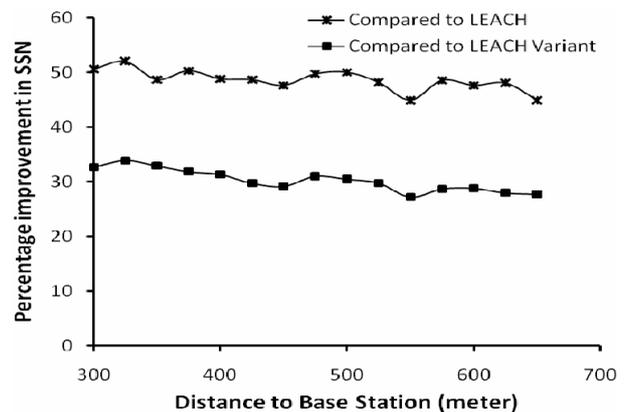


Figure 13. Improvement in network stability period in terms of First Node Dies (FND) in SSN for different positions of base station.

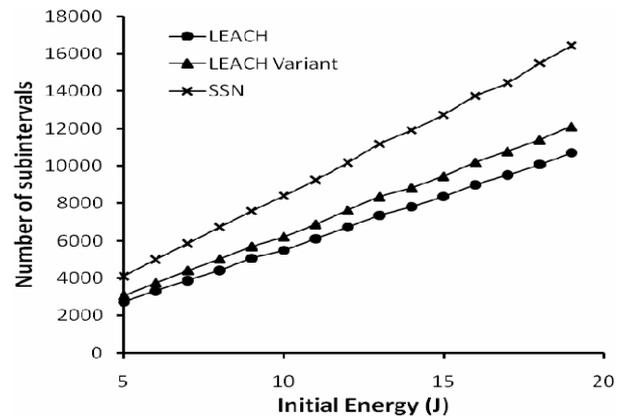


Figure 14. Network stability period in terms of First Node Dies (FND) for different initial energies of a sensor node.

varying this initial energy. We change the initial energy of a sensor node from 5 Joule with step of 1 Joule in each simulation run. We plot the values of network stability periods in terms of FND in **Figure 14** and the values of

HND in **Figure 15**.

Figure 15 indicates that HNDs of SSN, LEACH and LEACH variant are comparable. However, there is a significant steady improvement in FND for SSN over LEACH and its variant. We plot these improvements in **Figure 16**. The average improvement over LEACH and its variant is 52.04% and 34.25% accordingly.

These values clearly indicate that, SSN provides significantly higher time before first node dies in comparison to LEACH and its variant irrespective of data rate of sensor node or position of base station or initial energy of a sensor node.

6. Conclusions

We propose a novel self-organizing and adaptive clustering protocol SSN in this paper. We use three heuristics

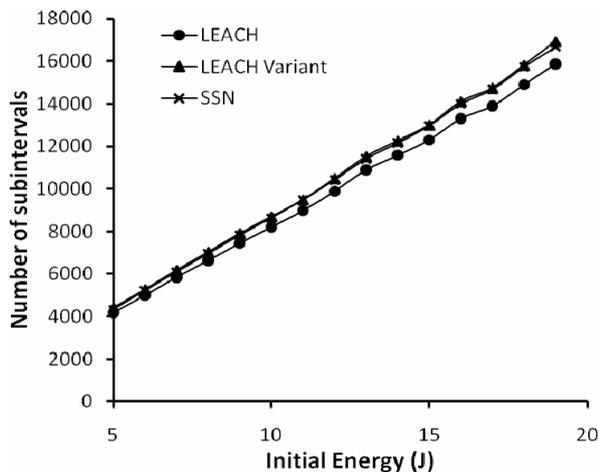


Figure 15. Half of the Nodes Die (HND) for different initial energies of a sensor node.

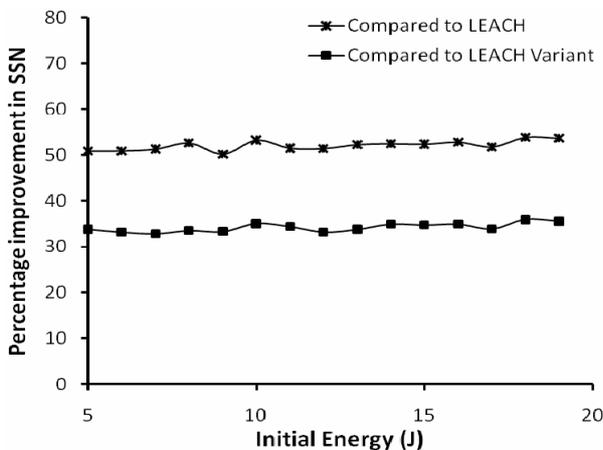


Figure 16. Improvement in network stability period in terms of First Node Dies (FND) in SSN for different initial energies of a sensor node.

for SSN with proper justifications. We present its complete mathematical formulation with the help of that of LEACH. We evaluate its stability period with that of LEACH and its best variant. We get a significant steady improvement in the evaluation in different circumstances.

7. Future Works

We propose SSN for homogeneous sensor nodes, *i.e.*, sensor nodes with similar transmission and sensing ranges. In our future works, we will attempt to enhance SSN for sensor nodes with different transmission and sensing ranges. We will also attempt to enhance SSN for multi radio sensor nodes, which are now emerging in the recent research works.

8. References

- [1] W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Microsensor Networks," *Proceedings of the Hawaii International Conference on System Sciences*, Maui, Vol. 2, January 2000, pp. 1-10.
- [2] J. Han and K. Micheline, "Data Mining: Concepts and Techniques," 2nd Edition, Morgan Kaufman, Elsevier, San Francisco, 2001.
- [3] S. Basagni, "Distributed Clustering Algorithm for Ad-Hoc Networks," *Proceedings of International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, Perth, 1999, pp. 310-315.
- [4] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks," *Proceedings of IEEE INFOCOM*, Anchorage, 2001, pp. 1028-1037.
- [5] M. Gerla, T. J. Kwon and G. Pei, "On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering," *Proceedings of IEEE WCNC*, Chicago, Vol. 1, 2000, pp. 100-105.
- [6] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 7, 1997, pp. 1265-1275.
- [7] J. Kamimura, N. Wakamiya and M. Murata, "Energy-Efficient Clustering Method for Data Gathering in Sensor Networks," *Proceedings of Workshop on Broadband Advanced Sensor Networks*, San Diego, Vol. 103, 2004, pp. 31-36.
- [8] J. Leu, M. H. Tsai, T. C. Chiang and H. Y. M. Huang, "Adaptive Power Aware Clustering and Multicasting Protocol for Mobile Ad Hoc Networks," *Lecture Notes in Computer Science*, Vol. 4159, September 2006, pp. 331-340.
- [9] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems," *IEEE Aerospace Conference Proceedings*, Big Sky, Vol. 3, 2002, pp. 1125-1130.
- [10] L. Li, S. Dong and X. Wen, "An Energy Efficient Clus-

- tering Routing Algorithm for Wireless Sensor Networks,” *Journal of China Universities of Posts and Telecommunications*, Vol. 3, No. 13, September 2006, pp. 71-75.
- [11] Y. Sangho, H. Junyoung, C. Yookun and H. Jiman, “PEACH: Power-Efficient and adaptive Clustering Hierarchy Protocol for Wireless Sensor Networks,” *Computer Communications*, Vol. 30, No. 14-15, 2007, pp. 2842-2852.
- [12] S. Ghiasi, A. Srivastava, X. Yang and M. Sarrafzadeh, “Optimal Energy Aware Clustering in Sensor Networks,” *Sensors Journal*, Vol. 2, No. 7, 2002, pp. 258-269.
- [13] H. Chan and A. Perrig, “ACE: An Emergent Algorithm for Highly Uniform Cluster Formation,” *Lecture Notes in Computer Science*, Springer Berlin/Heidelberg, Vol. 2920, February 2004, pp. 154-171.
- [14] O. Younis and S. Fahmy, “Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach,” *Proceedings of IEEE INFOCOM*, Hong Kong, Vol. 1, 2004, pp. 629-640.
- [15] J. Y. Cheng, S. J. Ruan, R. G. Cheng and T. T. Hsu, “PADCP: Poweraware Dynamic Clustering Protocol for Wireless Sensor Network,” *IFIP International Conference on Wireless and Optical Communications Networks*, Bangalore, 13 April 2006, pp. 1-6.
- [16] G. Smaragdakis, I. Matta and A. Bestavros, “SEP: A Stable Election Protocol for Clustered Heterogeneous Wireless Sensor Networks,” *Proceedings of Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA)*, Boston, 2004, pp. 251-261.
- [17] M. Haase and D. Timmermann, “Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection,” *4th International Workshop on Mobile and Wireless Communications Network*, San Jose, 2002, pp. 368-372.
- [18] W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, “An Application-Specific Protocol Architecture for Wireless Microsensor Networks,” *IEEE Transactions on Wireless Communications*, Vol. 14, No. 4, October 2002, pp. 660-670.
- [19] C. Nam, H. Jeong and D. Shin, “The Adaptive Cluster Head Selection in Wireless Sensor Networks,” *Proceedings of the IEEE International Workshop on Semantic Computing and Applications*, Incheon, 2008, pp. 147-149.
- [20] A. Giridhar and P. R. Kumar, “Maximizing the Functional Lifetime of Sensor Networks,” *Fourth International Symposium on Information Processing in Sensor Networks*, Los Angeles, 15 April 2005, pp. 5-12.
- [21] X. Wang, W. Gu, S. Chellappan, K. Schosek and D. Xuan, “Lifetime Optimization of Sensor Networks under Physical Attacks,” *Proceedings of IEEE International Conference on Communications (ICC)*, Seoul, Vol. 5, May 2005, pp. 3295-3301.
- [22] H. Zhang and J. C. Hou, “Maximizing α -Lifetime for Wireless Sensor Networks,” *3rd International Workshop on Measurement, Modeling, and Performance Analysis of Wireless Sensor Networks*, San Diego, 21 July 2005, pp. 70-77.
- [23] V. Rai and R. N. Mahapatra, “Lifetime Modeling of a Sensor Network,” *Proceedings of Design, Automation and Test in Europe*, Munich, Vol. 1, 7-11 March 2005, pp. 202-203.
- [24] M. U. Ilyas and H. Radha, “Increasing Network Lifetime of an IEEE 802.15.4 Wireless Sensor Network by Energy Efficient Routing,” *IEEE International Conference on Communications*, Istanbul, Vol. 9, June 2006, pp. 3978-3983.
- [25] L. Shi, A. Capponi, K. H. Johansson and R. M. Murray, “Sensor Network Lifetime Maximization Via Sensor Trees Construction and Scheduling,” *Third International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks*, Annapolis, June 2008.
- [26] P. Berman, G. Calinescu, C. Shah and A. Zelikovsky, “Power Efficient Monitoring Management in Sensor Networks,” *IEEE Wireless Communication and Networking Conference*, Atlanta, March 2004, pp. 2329-2334.
- [27] D. Brinza and A. Zelikovsky, “DEEPS: Deterministic Energy-Efficient Protocol for Sensor Networks,” *2nd ACIS International Workshop on Self-Assembling Wireless Networks*, Las Vegas, 2006, pp. 261-266.
- [28] A. Aung, “Distributed Algorithms for Improving Wireless Sensor Network Lifetime with Adjustable Sensing Range,” M.S. Thesis, Georgia State University, 2007.
- [29] S. G. Akojwar and R. M. Patrikar, “Improving Life Time of Wireless Sensor Networks Using Neural Network Based Classification Techniques with Cooperative Routing,” *International Journal of Communications*, Vol. 1, No. 2, 2008, pp. 75-86.
- [30] Y. Chen, Q. Zhao, V. Krishnamurthy and D. Djonin, “Transmission Scheduling For Sensor Network Lifetime Maximization: A Shortest Path Bandit Formulation,” *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, May 2006, pp. 145-148.
- [31] K. Dasgupta, K. Kalpakis and P. Namjoshi, “Improving the Lifetime of Sensor Networks via Intelligent Selection of Data Aggregation Trees,” *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2003, pp. 19-23.
- [32] H. Kang and X. Li, “Power-Aware Sensor Selection in Wireless Sensor Networks,” *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN)*, 2006.
- [33] M. Cardei and D. Du, “Summary on Improving Wireless Sensor Network Lifetime through Power Aware Organization,” *Seminar on Theoretical Computer Science, Wireless Networks*, Vol. 11, No. 3, 2005, pp. 333-340.
- [34] J. Park and S. Sahni, “An Online Heuristic for Maximum Lifetime Routing in Wireless Sensor Networks,” *IEEE Transactions on Computers*, Vol. 55, No. 8, August 2006, pp. 1048-1056.
- [35] J. Chang and L. Tassiulas, “Maximum Lifetime Routing

- in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, Vol. 12, No. 4, 2004, pp. 609-619.
- [36] P. Djukic and S. Valaee, "Maximum Network Lifetime in Fault Tolerant Sensor Networks," *IEEE Global Telecommunications Conference, GLOBECOM' 05*, St. Louis, Vol. 5, December 2005, pp. 3106-3011.
- [37] Y. T. Hou, Y. Shi, H. D. Sherali and S. F. Midkiff, "Prolonging Sensor Network Lifetime with Energy Provisioning and Relay Node Placement," *2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, September 2005, pp. 295-304.
- [38] M. Cardei, J. Wu, M. Lu and M. O. Pervaiz, "Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges," *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Montreal, Vol. 3, August 2005, pp. 438-445.
- [39] K. Kalpakis, K. Dasgupta and P. Namjoshi, "Efficient Algorithms for Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks," *Computer Networks*, Vol. 42, No. 6, 2003, pp. 697-716.
- [40] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, Vol. 5, No. 8, August 2006, pp. 2185-2193.
- [41] J. C. Choi and C. W. Lee, "Energy Modeling for the Cluster-based Sensor Networks," *Proceedings of the 6th IEEE International Conference on Computer and Information Technology*, Seoul, September 2006, p. 218.
- [42] S. Selvakenedy and S. Sinnappan, "An Energy-Efficient Clustering Algorithm for Multihop Data Gathering in Wireless Sensor Networks," *Journal of Computers*, Vol. 1, No. 1, April 2006, pp. 40-47.
- [43] A. B. M. A. A. Islam, "A Novel Approach To Cluster Heterogeneous Sensor Network (CHSN)," *MSc Engineering Thesis*, Vol. 4, May 2009, pp. 31-41.
- [44] A. B. M. A. A. Islam, C. S. Hyder, M. H. Kabir and M. Naznin, "Finding the Optimal Percentage of Cluster Heads from a New and Complete Mathematical Model on LEACH," *Wireless Sensor Network*, Vol. 2, No. 2, February 2010, pp. 129-140.
- [45] W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, 2002, pp. 660-670.
- [46] D. Song, "Probabilistic Modeling of Leach Protocol and Computing Sensor Energy Consumption Rate in Sensor Networks," Technical Report, Texas A & M University, February 2005.
- [47] T. Murata and H. Ishibuchi, "Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems," *Proceedings of 1st IEEE Conference Evolutionary Computation*, Orlando, Vol. 2, June 1994, pp. 812-817.
- [48] G. E. P. Box and M. E. Muller, "A Note on the Generation of Random Normal Deviates," *Annals of Mathematical Statistics*, Vol. 29, No. 2, 1958, pp. 610-611.